

Reduced-Order Modeling of Three-Dimensional External Aerodynamic Flows

Diego Alonso¹; José M. Vega²; Ángel Velázquez³; and Valentín de Pablo⁴

Abstract: A method is presented to construct computationally efficient reduced-order models (ROMs) of three-dimensional aerodynamic flows around commercial aircraft components. The method is based on the proper orthogonal decomposition (POD) of a set of steady snapshots, which are calculated using an industrial solver based on some Reynolds averaged Navier-Stokes (RANS) equations. The POD-mode amplitudes are calculated by minimizing a residual defined from the Euler equations, even though the snapshots themselves are calculated from viscous equations. This makes the ROM independent of the peculiarities of the solver used to calculate the snapshots. Also, both the POD modes and the residual are calculated using points in the computational mesh that are concentrated in a close vicinity of the aircraft, which constitute a much smaller number than the total number of mesh points. Despite these simplifications, the method provides quite good approximations of the flow variables distributions in the whole computational domain, including the boundary layer attached to the aircraft surface and the wake. Thus, the method is both robust and computationally efficient, which is checked considering the aerodynamic flow around a horizontal tail plane, in the transonic range $0.4 \leq \text{Mach number} \leq 0.8$, $-3^\circ \leq \text{angle of attack} \leq 3^\circ$. DOI: [10.1061/\(ASCE\)AS.1943-5525.0000148](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000148). © 2012 American Society of Civil Engineers.

Introduction

Aerodynamic design and certification are crucial steps in product development in various industrial sectors, including both mature (e.g., aeronautics and automotive) and emergent (e.g., wind turbine) industries. Current methods used to perform these tasks are largely based on low-fidelity simulation, ad hoc modelization, and wind tunnel tests. The former two provide rough approximations only, which are tested by the latter. Improved designs and reduced cost and time to market are increasingly needed to survive in increasingly competitive global market scenarios. Thus, a trend has been observed to substitute wind tunnel tests with computational fluid dynamics (CFD) simulations. The main difficulty is that the Reynolds number is quite large in actual industrial flows, which makes direct numerical simulation based on the exact compressible Navier-Stokes (NS) equations well beyond present (and predictable) supercomputer capability. Instead, turbulence models are widely used in industry, but these still require huge computational resources and CPU time. For instance, calculating one aerodynamic flow field around a commercial aircraft using a Reynolds averaged Navier-Stokes (RANS) solver requires about two CPU days in a cluster of

PCs, which makes CFD impractical, especially in multiparameter problems. In fact, up to seven parameters (angle of attack, sideslip angle, Reynolds and Mach numbers, and deflection angle of the various flight control surfaces) must be accounted for in flight envelope calculations, and tens of them are needed in shape optimization. Thus, reducing the computational cost of the CFD solvers is becoming the key step to facilitate their industrial use. Reduced order models (ROMs) are good candidates to provide reasonable approximations with a reasonable computational cost.

Among these, those based on proper orthogonal decomposition (POD) have been developed during the last 20 years because they drastically reduce the computational effort and allow for treating complex geometries. These models consist of (1) CFD-computing some flow snapshots, (2) extracting the most energetic POD modes from the snapshots, and (3) projecting the governing equations onto the resulting POD manifold. These models have been developed mainly for unsteady flows that are either linear or weakly nonlinear perturbations of base flows, or they only deal with time periodic flows, intending to cope with stability and control issues. Some examples can be found in the work by Dowell and Hall (2001), Rempfer (2003), Lucia et al. (2004), Lieu et al. (2006), and Thomas et al. (2010); the extension of these to general time-dependent, fully nonlinear flows in realistic industrial conditions has not been performed to our knowledge. Fully nonlinear, unsteady ROMs of simpler problems have received considerable attention in recent years (Couplet et al. 2005; Sirisup and Karniadakis 2005; Sirisup et al. 2005; Rapun and Vega 2010), but these are still several steps behind their efficient industrial use. Steady aerodynamic flows at large Reynolds number result from turbulence modeling when using RANS equations. Reduced order models for these flows in multiparameter problems are intrinsically nonlinear. The snapshots are themselves steady states for some selected parameter values, which must correspond to a coarse mesh in the parameter space if the required number of snapshots is kept within reasonable bounds. The ROM calculates flow

¹Doctor of Aerospace Engineering, E.T.S.I. Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, 28040 Madrid, Spain.

²Professor of Applied Mathematics, E.T.S.I. Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, 28040 Madrid, Spain.

³Professor of Aerospace Engineering, E.T.S.I. Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, 28040 Madrid, Spain (corresponding author). E-mail: angel.velazquez@upm.es

⁴Master of Aerospace Engineering, Airbus Operations, Paseo de John Lennon s/n. Getafe, 28906 Madrid, Spain.

Note. This manuscript was submitted on December 13, 2010; approved on June 24, 2011; published online on June 27, 2011. Discussion period open until March 1, 2013; separate discussions must be submitted for individual papers. This paper is part of the *Journal of Aerospace Engineering*, Vol. 25, No. 4, October 1, 2012. ©ASCE, ISSN 0893-1321/2012/4-588-599/\$25.00.

configurations for any set of parameter values, which need not be close to any of the CFD-calculated snapshots.

To our knowledge, the effort to derive fully nonlinear, steady industrial ROMs has focused in the two-dimensional (2D) case. LeGresley and Alonso (2001) derived a ROM for the strictly inviscid aerodynamic flow around a 2D airfoil, considering only perturbations of a base state as a result of shape bump functions. In particular, both the snapshots and the POD amplitudes were calculated using the Euler equations. More recently, Alonso et al. (2010) calculated POD modes using a RANS approximation but determined the POD-mode amplitudes projecting the Euler equations. To our knowledge, this is the first paper in the literature in which a ROM based on Euler equations has been derived that uses snapshots resulting from viscous CFD calculations, which has both conceptual and practical consequences. Alonso et al. (2010) considered a wide parameter range ($0.4 \leq M \leq 0.8$, $-3^\circ \leq \text{AoA} \leq 3^\circ$) in the Mach-number/angle-of-attack plane. A special shock wave treatment was necessary because of the presence of shock waves that move up to one-third of the chord as the parameters are varied. Moving shock waves lead to great difficulties when applying POD for one simple reason: POD reconstructions involve linear combinations of the snapshots, and the linear combination of shifted jumps does not yield a jump, but a stair-shaped array of jumps. If, instead, the shock wave does not move much as the parameters are varied, POD works quite well. Fortunately, this is the case with the optimized geometries in actual aircraft configurations of, for example, the fuselage, wings, and horizontal and vertical tail planes, provided that neither the Mach number nor the angle of attack is too high. Larger values of these could require a specific treatment, as shown in the Results section. Even if no moving shock wave is present, some additional difficulties can still be encountered, as shown in the following list:

- A. Industrial solvers contain unphysical terms, resulting from both turbulence modeling and the necessity to avoid numerical instability. Doubt naturally arises about what equations should be projected onto the POD manifold. The two obvious possibilities are the exact compressible NS equations and the equations that are actually used by the CFD solver, with the various unphysical artifacts included. Consistency of the results requires that both approaches yield the same result within the precision being sought.
- B. Even though industrial CFD meshes are fairly coarse compared with what would be required by direct numerical simulation, the number of mesh points is still huge in realistic industrial configurations. This leads to a great computational effort to project the governing equations onto the POD modes, at least if standard projection is made (defined through either volume or surface integrals), which involves all mesh points.

Coping with these two difficulties and providing ideas to derive robust and computationally efficient ROMs is the main object of this paper. The first difficulty (A) will deal with projecting the Euler equations onto the POD manifold. As explained and checked by Alonso et al. (2010), the use of the Euler equations to project the equations, while the original problem is turbulent (and the snapshots have been CFD-computed using the NS equations with a turbulence model and, possibly, some small stabilizing terms added for numerical reasons) relies on three related arguments:

- i. At a high Reynolds number, viscous effects are confined to thin and well-localized regions (boundary layers, shear layers, and shock waves), and thus contributions of the neglected viscous and stabilizing terms on the residual that is to be minimized [defined as averaging over the spatial domain; see Eq. (14)] are small.

- ii. The required vorticity distribution information needed to reconstruct the correct solutions (which cannot be completely calculated using Euler equations only) is already present in the POD modes.
- iii. The actual physics of the problem does not depend on the turbulence model and the stabilizing terms used in CFD; the latter are only numerical artifacts.

Projecting the Euler equations is quite convenient, envisaging practical industrial applications:

- a. The resulting ROM is independent of the CFD solver. Such flexibility facilitates the calculation of the snapshots, which is the most computationally expensive part of the process. In particular, different solvers, based on different turbulence models, can be used simultaneously.
- b. Using the Euler equations reduces the order of the spatial derivatives involved in the projection step, which is quite convenient to minimize the effect of discretization errors.
- c. The ROM itself does not need artificial viscosity terms to stabilize the convergence process. Furthermore, even if the CFD flow solver requires artificial terms (which is most likely), the ROM does not mimic these terms. Thus, it produces what could be called a clean solution.

Projection onto the POD manifold will be done minimizing a properly defined, positive residual calculated from the Euler equations and the boundary conditions. To isolate the construction of the ROM from the minimization process, a genetic algorithm will be used to accomplish the latter. This will have the additional advantage of requiring a fast calculation of the residual to compensate the low computational efficiency of the genetic algorithm. A further computational efficiency improvement would result from using gradient-like minimization methods, but these will not be applied in this paper.

The second difficulty (B) will be dealt with using residuals defined out of a limited number of mesh points, much smaller than the total number of points in the computational mesh. Moreover, the selected points can be either scattered along the whole flow domain or concentrated in a smaller projection window. The selection of both the projection window and the points inside the window is not critical. Instead, it is subject to only mild limitations, which have some additional advantages:

- B-1. A judicious selection of the projection window improves the quality of the POD modes and reduces its number for a given precision. Regions where the flow shows the most significant features are the best candidates.
- B-2. Regions with concentrated viscous effects (where the Euler equations do not apply) and large CFD concentrated errors can be both avoided.

The resulting approach has some (somewhat) unexpectedly good outcomes. For instance, even though the projection onto the POD manifold is based in the Euler equations and neither the wake nor the viscous boundary layer attached to the wall is used in the ROM calculations, the velocity profile inside the wake and the boundary layer are both reconstructed well. Furthermore, the velocity profiles in the whole computational mesh are well reconstructed, too. All these will appear as reasonable after a closer look at the basic ideas behind the process and will also suggest further improvement, that is, that the POD modes can be calculated from a local set of snapshots in the parameter space. For the sake of illustration, a method will be applied to reconstruct the aerodynamic flow around a horizontal tail plane (HTP).

The remainder of the paper is organized as follows. The CFD method to calculate the snapshots will be briefly described next, followed by the development of the ROM and its application to the aforementioned aerodynamic flow around a HTP. Some

concluding remarks comment on the scope and possible extensions of the paper.

CFD Calculation of the Snapshots and POD Modes

The derivation of the ROM will be illustrated and the ROM itself tested, considering the aerodynamic flow around a three-dimensional (3D) HTP [Fig. 1(a)]. Snapshots will be calculated using the ELSA code (Cambier and Gazaix 2002; Ben Kheli et al. 2002), developed by ONERA and CERFACS, which is a finite volume discretization (Tannehill et al. 1997) of the compressible continuity, momentum, and energy equations, with viscous terms modified according to an Edwards-corrected (Edwards and Chandra 1996) Spalart-Almaras turbulence model (Spalart and Allmaras 1992) and some extra stabilizing terms added for numerical reasons. Further details of the numerical solver itself will not be needed to develop the ROM.

The HTP has a span length 1.5 times its root chord, c , and a thickness at the root section equal to $0.115c$. The Cartesian coordinate frame is centered at the leading edge point of the root section, with the x - z plane containing the root section itself. The x -axis passes through the trailing edge of the root section, the z -axis points toward the suction side, and the y -axis points toward the HTP tip [Fig. 1(b)]. The complete computational mesh [Fig. 1(b)] is block structured, with 48 blocks and a total number of 3,053,744 mesh points in a parallelepipedic computational domain of sides $-5c < x < 10c$, $0 < y < 10c$, and $-10c < z < 10c$. Those blocks that are adjacent to the HTP surface make up what will be called the O-mesh [also appreciated as the shaded region that surrounds the root section in Fig. 1(b)], whose wall-normal size is approximately one-third of the HTP thickness; the volume of the O-mesh is 3.7×10^{-6} times the volume of the computational domain. Mesh points concentrate in the O-mesh, which contains 840,825 mesh points. In fact, mesh points concentrate near the planes $x = 0$, $y = 0$, and $z = 0$ in regions whose widths are comparable to the chord, the span, and the thickness of the HTP, respectively [as shown in the shadowed regions in Fig. 1(a)]. Such concentration also facilitates topological compatibility among blocks. In fact, the ROM will be constructed using only a part of the O-mesh (both to calculate the POD modes and to project the governing equations) and some points in the $x = -5c$ plane (to impose the upstream boundary conditions).

The original flow variables are the three velocity components, u , v , and w (along the x -, y -, and z -axes, respectively), the pressure p , the density ρ , and the temperature T . However, instead of using these variables, the equation of state for ideal gases is substituted into the remaining equations, and the resulting five state variables are redefined as the mass fluxes in the x , y , and z directions, which are ρu , ρv , and ρw , respectively, and the density and the pressure, which are all nondimensionalized using their free stream values. Because the sideslip angle is zero, the upstream boundary conditions, BC (which are imposed at the plane $x = -5c$), are

$$\begin{aligned} BC_1 &\equiv \rho u - M \cos \text{AoA} = 0, & BC_2 &\equiv \rho w - M \sin \text{AoA} = 0, \\ BC_3 &\equiv \rho v = 0, & BC_4 &\equiv \rho - 1 = 0, & BC_5 &\equiv p - 1 = 0 \end{aligned} \quad (1)$$

The parameter space is a rectangle in the AoA - M (angle-of-attack versus Mach-number) plane, in the range $-3^\circ \leq \text{AoA} \leq 3^\circ$ and $0.4 \leq M \leq 0.8$ (see Fig. 2). The flow around the HTP in such a range presents strong shock waves, but these move only slightly as the

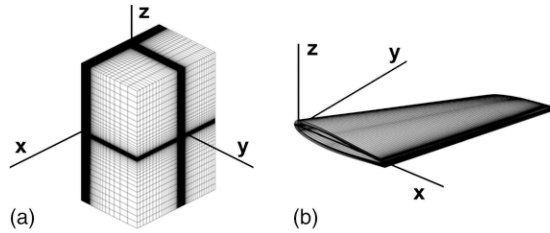


Fig. 1. Three-dimensional view of (a) the whole computational mesh and (b) blow-up of the HTP with the O-mesh attached

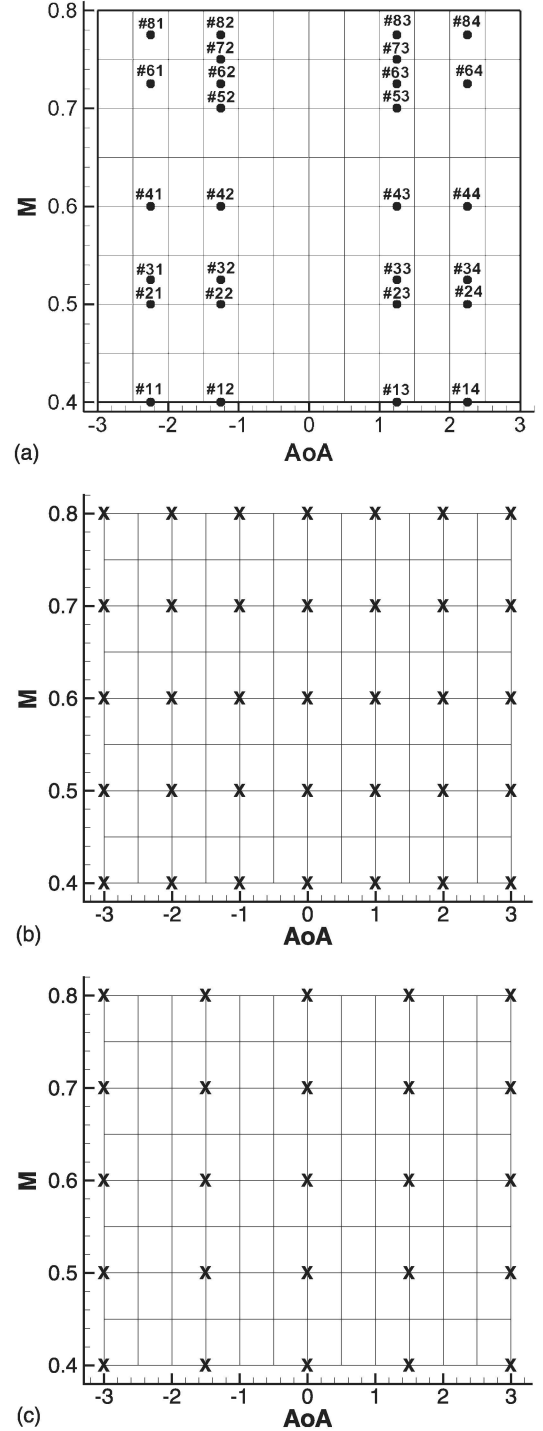


Fig. 2. (a) Test points (filled circles) and the snapshots in Combination 1 (crossing points in the plotted grid); the snapshots (crosses) in (b) Combinations 2 and (c) 3

parameters are varied, unlike the ones in the 2D case studied by Alonso et al. (2010). This means that no special shock wave treatment is needed in this case.

The snapshots are steady distributions of the flow variables for parameter values that should be representative of the parameter range that is being considered. These parameter values and the associated snapshots are denoted as

$$(AoA_k, \mathbf{M}_k) \quad \text{and} \quad \mathbf{q}_k = [(\rho u)_k, (\rho v)_k, (\rho w)_k, p_k, p_k] \\ \text{for } k = 1, \dots, N_0 \quad (2)$$

The POD modes resulting from these snapshots, $\mathbf{Q}_1, \dots, \mathbf{Q}_{N_0}$, and the reconstruction of the snapshots in terms of these modes are both given by the standard formulas

$$\mathbf{Q}_j = \sum_{k=1}^{N_0} \alpha_j^k \mathbf{q}_k, \quad \mathbf{q}_k = \sum_{j=1}^{N_0} \alpha_j^k \mathbf{Q}_j \quad (3)$$

The coefficients $\alpha_1^k, \dots, \alpha_{N_0}^k$ are the eigenvectors of the positive definite, symmetric $(N_0 \times N_0)$ -matrix \mathbf{R} , known as the covariance matrix, whose elements are given by

$$R_{ij} = \langle \mathbf{q}_i, \mathbf{q}_j \rangle \quad (4)$$

in terms of an inner product that will be defined in Eqs. (15) and (17).

Now, if the second expansion [Eq. (3)] is truncated to $n \leq N_0$ terms, then the root mean square (RMS), relative error in reconstructing all snapshots is given by

$$\text{RMS error} = \sqrt{\frac{\sum_{i=n+1}^{N_0} \gamma_i}{\sum_{i=1}^{N_0} \gamma_i}} \quad (5)$$

where $\gamma_1 \geq \dots \geq \gamma_{N_0} \geq 0$ are the eigenvalues of the matrix in Eq. (4), namely, the squares of the POD singular values. Here, the RMS error is defined in terms of the norm associated with the inner product that is used to define the covariance matrix in Eq. (4). Equation (5) gives an a priori estimate of the number of POD modes that should be retained to obtain an approximation of the flow variables in a generic point of the parameter space within a prescribed error. Of course, the required number of POD modes (and the quality of these) depends on how representative the selected set of snapshots is. The snapshots will be selected as those that are closest (in an appropriate sense) to the flow configuration that is being calculated. But such selection will be made among a set of snapshots that are equispaced in the parameter space, since a better selection of the snapshots location in the parameter space is outside the scope of this paper.

The flow variables are written as expansions in the POD modes, as follows:

$$\mathbf{q}(x, y, z; AoA, \mathbf{M}) = \sum_{j=1}^n A_j(AoA, \mathbf{M}) \mathbf{Q}_j(x, y, z) \quad (6)$$

where the POD-mode amplitudes A_j = the unknowns that will be determined by the ROM. Note that these amplitudes can be calculated for the parameter values associated with the snapshots using the second expression in Eq. (3), truncated to n terms, as

$$A_j(AoA_k, \mathbf{M}_k) = \alpha_j^k \quad (7)$$

This allows for obtaining a first approximation of the POD-mode amplitudes at any point of the parameter space using interpolation in each amplitude, in the AoA versus \mathbf{M} plane. Here, we use the

Akima's bivariate interpolation method for scattered data (Akima 1978) on the amplitudes A_j .

Derivation of the Reduced-Order Model

The basic idea to construct a ROM that provides the POD amplitudes at any point in the parameter space consists of minimizing a positive definite residual of the governing equations and boundary conditions. Such residual \mathcal{H} will be a positive function of the flow variables' distributions that vanishes only when both the equations and boundary conditions are identically satisfied. The POD-mode amplitudes can be obtained substituting the expansion [Eq. (6)] into the residual \mathcal{H} and minimizing the function $\mathcal{H} = \mathcal{H}(A_1, \dots, A_n)$.

Proceeding in this way allows for several possible improvements, which will be dealt with in the following sections. In particular, the following issues will be addressed:

1. Deciding what equations (either the exact equations or those that are used by the CFD solver) should be used in the definition of the residual. Related simplifications deal with defining the residual in terms of the Euler equations and using these in conservative form.
2. Selecting the inner product that is most convenient in the definition of the covariance matrix, to calculate the POD manifold.
3. Selecting those points in the computational mesh that will be used to define the residual. In fact, it will be seen that considering only a part of the O-mesh as projection window is enough to both appropriately calculate the POD modes and define the residual.
4. Dividing the computational domain into subdomains and applying the method to each subdomain separately. Such subdivision is convenient when the subdomains behave somewhat independently among each other.
5. Defining the POD manifold locally in the parameter space.

Using the Euler Equations in Conservative Form to Calculate the Residual

The first question is which governing equations should be used to define the residual. These can be the exact governing equations or the approximations that are implicit in the CFD solver (i.e., the RANS equations with stabilizing terms). If the latter equations are used, then the resulting ROM will depend on the CFD solver. Instead, we will ignore CFD details and take advantage of the fact that the Reynolds number is large to define the residual in terms of the Euler equations. Still, since numerical calculation of spatial derivatives is less precise than calculation of the flow variables themselves, the conservative form of the Euler equations will be used, and the divergence theorem will be applied. Thus, the residual will be defined in terms of surface integrals that (in principle) should be extended to all elementary parallelepipedic cells in the computational mesh. Each of these is bounded by eight neighboring mesh points.

Note that the ROM is intended to provide (within an approximation comparable to that of the RANS equations) the correct vorticity distribution along the flow field, which is in contrast with ROMs based on snapshots calculated by the Euler equations. Here, the vorticity distributions are already contained in the RANS calculated snapshots. As already noticed by us in a related problem (Alonso et al. 2010), the justification of the combined use of the RANS and Euler equations is made in two steps:

1. RANS equations differ from the exact equations in the turbulence modeling terms and the numerical stabilizers, which are expected to have a small effect on the larger scales. These are accounted for by the most energetic POD modes. Thus, if

only the latter are retained, the effect on the unphysical terms in the RANS equations should be small, and the RANS equations can be replaced by the exact equations.

2. Both calculating the POD modes and projecting the exact equations onto the POD manifold is made adding the pointwise values of the flow variables distributions/governing equations for many points (say, on the order of 1,000; see following section) in the computational mesh, which involves a spatial averaging. Thus, if the Reynolds number is large, the effect of viscous and thermal conduction terms in both the calculation of the covariance matrix and the residual is small because viscous effects are localized in small spatial regions (namely, the boundary layers, shear layers, and shock waves).

A further increase of the computational efficiency will result from substituting the pointwise Euler equations by the following conservation equations, which are obtained by integrating the Euler equations (written in conservative form) in a generic domain Ω and applying the divergence theorem

$$EQ_1(\Gamma) \equiv \frac{1}{\mathcal{A}} \iint_{\Gamma} (\rho u n_x + \rho v n_y + \rho w n_z) dS = 0 \quad (8)$$

$$EQ_2(\Gamma) \equiv \frac{1}{\mathcal{A}} \iint_{\Gamma} \left\{ \left[\frac{(\rho u)^2}{\rho} + \frac{p}{\gamma} \right] n_x + \left(\frac{\rho u \rho v}{\rho} \right) n_y + \left(\frac{\rho u \rho w}{\rho} \right) n_z \right\} dS = 0 \quad (9)$$

$$EQ_3(\Gamma) \equiv \frac{1}{\mathcal{A}} \iint_{\Gamma} \left\{ \left(\frac{\rho v \rho u}{\rho} \right) n_x + \left[\frac{(\rho v)^2}{\rho} + \frac{p}{\gamma} \right] n_y + \left(\frac{\rho v \rho w}{\rho} \right) n_z \right\} dS = 0 \quad (10)$$

$$EQ_4(\Gamma) \equiv \frac{1}{\mathcal{A}} \iint_{\Gamma} \left\{ \left(\frac{\rho w \rho u}{\rho} \right) n_x + \left(\frac{\rho w \rho v}{\rho} \right) n_y + \left[\frac{(\rho w)^2}{\rho} + \frac{p}{\gamma} \right] n_z \right\} dS = 0 \quad (11)$$

$$EQ_5(\Gamma) \equiv \frac{1}{\mathcal{A}} \iint_{\Gamma} e (\rho u n_x + \rho v n_y + \rho w n_z) dS = 0 \quad (12)$$

where γ = ratio of specific heats, Γ = boundary of the domain Ω , \mathcal{A} = area of Γ , $\mathbf{n} = (n_x, n_y, n_z)$ = outward unit normal to Γ , and e = modified mechanical energy per unit mass, defined as

$$e = \rho^{-2} \left\{ \rho p + \left[\frac{(\gamma - 1)}{2} \right] [(\rho u)^2 + (\rho v)^2 + (\rho w)^2] \right\} \quad (13)$$

Equations (8)–(12) apply in any smooth surface Γ contained in the computational domain; and conversely, imposing these equations in the boundaries of all elementary cells in the computational domain provides a good (finite volume) approximation of the solution. Using this, the residual is defined as

$$\mathcal{H} = \sum_{k=1}^{N_E} \sum_{i=1}^5 \sqrt{|EQ_i(\Gamma_k)|} + \sum_{m=1}^{N_{BC}} \sum_{i=1}^5 \sqrt{|BC_i(x_m, y_m, z_m)|} \quad (14)$$

where BC_i and EQ_i are as defined in Eqs. (1) and (8)–(12), respectively, and in principle, the sums are extended to the N_E elementary cells (each bounded by eight neighboring points in the computational mesh) and the N_{BC} points in the boundary of the computational mesh; but the residual will be calculated using only a limited number of mesh points, as it will be explained subsequently. This will be (modulo some additional improvements that will be introduced in the following subsections) the residual that will be minimized to obtain the reduced order model.

If the expressions inside the square roots in Eq. (14) were squared, then a definition more alike to the L_2 norm would be obtained, but such an expression would be more sensitive to CFD localized errors, which can be somewhat large. This has been discussed by us (Alonso et al. 2009, 2010) in a related incompressible fluid dynamics problem, comparing various possible definitions of the residual. Here, we just note that if the expressions inside the square roots in Eq. (14) were squared and the typical CFD errors were on the order of, say, 10^{-3} , but localized errors in a $O(10^{-2})$ region of the computational domain were on the order of, say, 10^{-1} , then when both were squared and added up, localized errors would mask the relevant errors resulting from how well the equations are satisfied, which are the ones that should be minimized by the method. The result would be that the POD-mode amplitudes would be selected in the minimization process to minimize errors where CFD errors are concentrated, giving a spurious solution. This will not happen when the residual Eq. (14) is minimized. All these mean that a good selection of the residual should be calibrated taking into account localized CFD errors, which could require in some cases replacing the square root in Eq. (14) by a p th order root, with p selected after some calibration; the value $p = 2$ used here has proven to be a good selection also in related problems (Alonso et al. 2009). On the other hand, the effect of those localized errors whose location is known a priori can also be avoided using an appropriate projection window, as defined in the next section.

Selection of the Inner Product to Calculate POD Modes

In principle, the inner product to define the covariance matrix Eq. (4) could be the standard L_2 inner product, namely

$$\langle \mathbf{q}_i, \mathbf{q}_j \rangle_{L_2} = \iiint_{\Omega} \mathbf{q}_i \cdot \mathbf{q}_j \, dx \, dy \, dz \quad (15)$$

where \cdot = the standard vector inner product on the flow state vector \mathbf{q} , namely [cf. Eq. (2)],

$$\mathbf{q}_i \cdot \mathbf{q}_j = (\rho u)_i (\rho u)_j + (\rho v)_i (\rho v)_j + (\rho w)_i (\rho w)_j + \rho_i p_j + p_i \rho_j \quad (16)$$

Instead, we can use the following inner product, which is somewhat consistent with the residual Eq. (14), namely

$$\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \sum_{k=1}^{N_E} \left(\iint_{\Gamma_k} \mathbf{q}_i \, dA \right) \cdot \left(\iint_{\Gamma_s} \mathbf{q}_j \, dA \right) \quad (17)$$

where Γ_k = boundaries of the N_E elementary cells in the computational domain.

The POD manifolds obtained using these two inner products are quite close to each other. But the inner product Eq. (17) is more favorable from the computational point of view, in terms of both the required CPU time and RAM. This is because the numbers of operations required to calculate the covariance matrix in Eq. (4)

using Eqs. (15) and (17) scale with $N_E N_0^2$ and $N_E N_0$, respectively, where N_0 is the number of snapshots. In addition, the L_2 inner product requires either (1) to read from the hard disk all snapshots (i.e., N_0 snapshots, which involve $N_0 N_E$ snapshots components) and store them at a time (which requires a storing space that exceeds the size of the RAM of a personal computer in typical aerodynamic applications) or (2) to store only two of them at a time to calculate their inner product, which must be done $N_0(N_0 - 1)/2$ times and requires a large amount of CPU time. Instead, the inner product [Eq. (17)] only requires to read and store each snapshot once [to calculate the surface integrals appearing in the right hand side of Eq. (17)]. Thus, the covariance matrix will be calculated using the inner product [Eq. (17)] in all applications in the next section.

Using a Limited Number of Elementary Cells in a Projection Window to Calculate Surface Integrals in the Residual and the Covariance Matrix

As anticipated in the preceding section, the POD-mode amplitudes are calculated minimizing the following function:

$$\mathcal{H} = \mathcal{H}(A_1, \dots, A_n) \quad (18)$$

which is obtained substituting the expansions Eq. (6) into the residual Eq. (14). Now, the most computationally expensive part of the process is the calculation of the residual, which involves a number of operations comparable to the total number of points ($\sim 3 \cdot 10^6$ in the subsequent HTP application). If the residual Eq. (14) was replaced by polynomial expressions in the flow variables (by, e.g., squaring the expressions inside the square roots), then the resulting residuals could be preprocessed to obtain polynomials in the POD-mode amplitudes, but the number of coefficients of such polynomials scales with n^q (a usually large number), where n = number of retained modes and $q \geq 4$ = degree of the polynomials.

A second way of reducing the computational cost of the process has been already checked in a related problem (Alonso et al. 2009) and arises from the observation that minimizing Eq. (18) can be seen as solving n equations. On the other hand, the residual involves flow information from N_E elementary cells, where N_E is much larger than n if all elementary cells are accounted for. In fact, the total number of elementary cells in the computational domain is selected by the CFD solver to fulfill numerical requirements, whereas the number of retained modes is related to the actual aerodynamic information that is present in the snapshots. Now, if all calculations were exact and the information in the elementary cells were nonredundant, the aforementioned n equations would be independent among each other, taking $N_E = n$. Because CFD calculations are not exact, the number of surface integrals in the residual Eq. (14) must be taken somewhat larger than n (say, $N_E = 3n$), but by no means equal to the much larger total number of elementary cells. Similarly, calculating POD modes does not require information from all mesh points, and thus the number of surface integrals in the inner product Eq. (17) will also be taken to be just somewhat larger than the number of retained modes.

The selected elementary cells to calculate both the covariance matrix and the residual can be either scattered over the computational domain or concentrated in a projection window. The second choice is generally better if the projection window is selected to include the relevant aerodynamic information, subject to only mild limitations:

1. Because the residual is based on the Euler equations, excluding regions where viscous effects are significant is a good choice. This can only be done for the boundary layers attached to solid walls, given that the location of the remaining regions

(e.g., shock waves and shear layers) is not known a priori for the flow configuration that is being calculated.

2. Excluding regions of large concentrated localized CFD errors is also a good choice to avoid their spurious effect.

In the application to the HTP in the next section, the O-mesh around the HTP surface will be taken as projection window, excluding both the boundary layer and a portion of the O-mesh near the HTP tip, where CFD errors are concentrated. The covariance matrix will be calculated using 208 elementary cells (scattered over the projection window) to calculate the surface integrals appearing in the expressions of both the residual [Eq. (14)] and the inner product [Eq. (17)]. This means that the computational effort has been divided by the ratio of the total number of elementary cells and the selected number of these, which is on the order of $4 \times 10^5 / 100 = 4,000$.

Similarly, instead of the N_{BC} points that are present in the boundary of the computational domain, the residual Eq. (14) can be calculated using a smaller number of points, concentrated in a part of the boundary. In the application to the HTP in the next section, only those points at the face $x = -5c$ of the boundary of the computational domain will be used to impose the boundary conditions.

Division of the Projection Window into Subdomains

As noticed and explained by Lorente (2009), the aerodynamic flow near an aircraft part may behave in an independent fashion (as the parameters are varied) in various subdomains of the computational domain. In this case, the number of POD modes needed to describe the flow field in the whole domain for a given accuracy is approximately equal to the product of the numbers of modes required to approximate it in the subdomains. Thus, dividing the projection window into subdomains may be convenient to reduce the effective number of POD modes. As an additional advantage, the resulting process could be easily parallelized because both calculation of the POD modes and minimization of the residual are carried out independently in the various subdomains.

Now, once we have selected the O-mesh to calculate both the POD modes and the residual in the HTP application, we note that the suction and pressure sides behave fairly independently of each other as the angle of attack and the Mach number are varied. Intuitively, it can be guessed that increasing the Mach number increases the velocity in both regions, but an increment in the angle of attack produces opposite trends in the pressure and suction sides. Thus, we consider two disjoint subdomains in the O-mesh that cover the suction and pressure sides. In addition, the residual is calculated in each subdomain using only 104 elementary cells instead of the 14,430 that are present. The interface between both subdomains is close to the stagnation point, whose surroundings bear quite important flow information and show steep gradients. This means that the reconstructions in both subdomains, where both POD modes and the residual are calculated (and the latter is minimized), may show discontinuities at the common boundary. The latter are smoothed out projecting the reconstructed flow distributions onto a set of POD modes calculated for the whole computational domain. Note that the latter projection also provides the reconstruction of all flow variables distributions in the whole computational domain, which will be checked in the Results section.

Using a Local POD Manifold

If POD modes are defined globally to cover the whole parameter space, then the number of these is larger than if POD modes are defined locally. In other words, the dimension of the POD manifold can be decreased using those snapshots that are closest to the

solution that is being sought. Doing that increases the computational effort associated with POD calculations. But the latter is quite inexpensive compared with the remaining ROM calculations, whose computational cost increases as the numbers of modes are increased. Thus, the overall computational cost is significantly decreased. Such local definition of the POD manifold to calculate the solution at a point (AoA , M) in the parameter plane is made in three steps.

A first approximation of the state vector, $\tilde{\mathbf{q}}$, is obtained using Shepard interpolation (Shepard 1968) of neighboring snapshots.

Now, a distance from each snapshot (labeled with the index j) to the approximation $\tilde{\mathbf{q}}$ is defined through the orthogonal projection, as

$$D_j = 1 - \frac{\langle \tilde{\mathbf{q}}, \mathbf{q}_j \rangle}{\left(\sqrt{\langle \tilde{\mathbf{q}}, \tilde{\mathbf{q}} \rangle} \sqrt{\langle \mathbf{q}_j, \mathbf{q}_j \rangle} \right)} \quad (19)$$

Here, the inner product is the same used to calculate the POD modes, namely, that defined in Eq. (17). Note that the local POD manifold is to be defined independently for each of the subdomains defined in the last subsection. The N_1 nearest snapshots [with the distance Eq. (19)] are considered, where N_1 must be somewhat large compared with the required number of POD modes (say, twice as many). Then the number of POD modes, \tilde{N}_1 , is selected using the a priori error estimate [Eq. (5)] to keep RMS errors within a specified bound ε_{LM} (which is to be chosen after some calibration).

As a last step, a safety factor $F > 1$ is defined and the $N_2 = F\tilde{N}_1$ nearest snapshots are retained to evaluate the POD manifold that is finally used to calculate the residual. Note that N_2 can be either smaller or larger than N_1 .

Once the local set of snapshots has been selected, the local POD manifold is constructed (in each of the subdomains that are considered, as explained previously), retaining the appropriate number of modes to keep the a priori error estimate [Eq. (5)] smaller than a required error-bound ε_{GA} .

Summarizing the ROM Derivation Method

The method to calculate the flow variables for given values of the parameters proceeds in seven steps:

1. To begin with, N_0 snapshots are CFD-calculated for representative values of the parameters in a given parameter range.
2. A projection window is defined and the location of some elementary cells is chosen. The latter are used to calculate the surface integrals appearing in the residual Eq. (14) and the inner product Eq. (17). The computational domain is divided into various subdomains.
3. For each subdomain, a local selection of snapshots is made using the algorithm already described. Such selection is performed for each subdomain independently, which allows for calculating POD modes independently in each subdomain. POD modes are calculated using the covariance matrix Eq. (4), with the inner product defined in Eq. (17).
4. The flow variables are expanded (in each subdomain) into POD modes as defined in Eq. (6). The POD-mode amplitudes A_i depend on the subdomain but are common to the five flow variables.
5. The number of retained modes within a prescribed accuracy ε_{GA} can be obtained using the error estimate Eq. (5). An initial guess for the amplitudes values is obtained via POD plus interpolation, as already explained.
6. The amplitudes are calculated in each subdomain minimizing the residual Eq. (14).

7. The state variables are reconstructed in each subdomain of the projection window using their associated sets of amplitudes. The flow field in the whole computational domain is obtained merging the solutions in the various subdomains as already explained, and projecting the resulting merged solution onto a set of most energetic global POD modes, obtained in the whole computational domain applying POD to the complete set of snapshots.

Results

Let us now check the ability of the ROM developed in last section to provide the aerodynamic flow around the HTP already described. The various parameters that have been left free in the description of the preceding method are chosen after some calibration for the HTP as follows:

- The upper bounds of the RMS error required to select the number of retained modes and the local set of snapshots are $\varepsilon_{GA} = \varepsilon_{LM} = 10^{-3}$; the initial guess of the number of local snapshots and the associated safety factor are $N_1 = 40$ and $F = 2.5$, respectively.
- The residual is minimized using a genetic algorithm, whose parameters are as follows. The total number of individuals is 10,000, with a discretization of 10 bits per *chromosome* (i.e., POD-mode amplitude) of each individual; the span allowed around the POD plus interpolation initial solution is equal to 50%, and 2% of the elite individuals go straight into the next generation. The crossover probability is equal to 0.8, and 5,000 bits are mutated in each generation. The process is completed if the residual remains constant during 100 generations.

Using the ELSA code, 117 snapshots are calculated at all combinations of 13 equispaced values of the angle of attack between -3 and 3° , and nine equispaced values of the Mach number between 0.4 and 0.8 [see Fig. 2(a)]. In addition, 28 test points [plotted in Fig. 2(a)] will be used to check the results. These test points are denoted as PTX_1X_2 , where $X_1 = 1, \dots, 8$ labels the following values of the Mach number $M = 0.4, 0.5, 0.525, 0.6, 0.7, 0.725, 0.75$, and 0.775 , and $X_2 = 1, \dots, 4$ labels the following values of the angle of attack $-2.25, -1.25, 1.25$, and 2.25° ; note that not all combinations of these values of M and AoA are considered. These test points have been selected as representative of the parameter space, with some emphasis on the transonic regime.

To test the effect of the number of snapshots in the quality of the results, three combinations of snapshots (see Fig. 2) are used to calculate the POD manifold:

- Combination 1 consists of the whole set of 117 calculated snapshots.
- Combination 2 contains the $(7 \times 5 =) 35$ snapshots resulting from all combinations of seven equispaced values of the angle of attack between -3 and 3° and five equispaced values of the Mach number between 0.4 and 0.8.
- Combination 3 contains the $(5 \times 5 =) 25$ snapshots resulting from all combinations of five equispaced values of AoA between -3 and 3° and five equispaced values of M between 0.4 and 0.8.

As explained previously, the ROM provides the distributions of all flow variables in the whole computational domain. Such distributions allow for calculating any property of the aerodynamic flow. The ability of the ROM to provide overall aerodynamic loads is checked considering the lift, pressure drag, and lateral force coefficients, defined in terms of surface integrals over the HTP surface, S_{HTP} , as

$$(C_L, C_{PD}, C_Y) = \iint_{S_{HTP}} (n_L, n_{PD}, n_Y) C_p dS \quad (20)$$

where $\mathbf{n} = (n_x, n_y, n_z)$ = outward unit wall normal to S_{HTP} , $n_L = -n_x \sin AoA + n_z \cos AoA$, $n_{PD} = n_x \cos AoA + n_z \sin AoA$, and C_p = pressure coefficient, defined as

$$C_p = 2 \left[\frac{(p - 1)}{\gamma M^2} \right] \quad (21)$$

Similarly, the roll, pitch, and yaw moment coefficients are defined as

$$(C_R, C_M, C_N) = - \iint_{S_{HTP}} [(\mathbf{x} - \mathbf{x}_0) \times \mathbf{n}] C_p dS \quad (22)$$

where $\mathbf{x} = (x, y, z)$ = position vector, and $\mathbf{x}_0 = (c/4, 0, 0)$ = reference point to calculate moments.

RMS and maximum errors of the reconstructed aerodynamic coefficients in the 28 test points selected are shown Table 1. The second and third columns show the errors resulting from calculating the covariance matrix using all elementary cells in the O-mesh and 208 of them; the residual is calculated using 208 elementary cells in both cases. The RMS errors for the remaining two combinations of snapshots are provided in the remaining two columns, as calculated using 208 elementary cells.

1. The errors resulting from using the whole O-mesh and 208 surface integrals are comparable, which confirms that using a limited number of points does not degrade precision.
2. The errors in the pressure drag, lateral force, and yaw moment coefficients are larger than the errors in the remaining coefficients, which is because the actual values of C_{PD} , C_Y , and C_N are smaller than the remaining coefficients.
3. Maximum errors are at most three times larger than their RMS counterparts and are concentrated at those test points that exhibit shock waves (see Figs. 3–5).
4. Combination 1 generally yields better results than Combination 2, and this provides smaller errors than Combination 3, which was to be expected according to the numbers of snapshots involved in the three combinations. However, it is remarkable that Combination 3 provides results that are good enough in industrial applications, using only 25 equispaced snapshots to cover the whole parameter range, which includes transonic conditions.
5. It is clear that if the snapshots were located in an appropriate way in the parameter space, then the results would further improve using a smaller number of snapshots. This selection could be done by hand, appropriately concentrating snapshots in the high-Mach-number/high-angle-of-attack regions. However, such an ad hoc selection is not practical from the industrial point of view. An automatic way of selecting the snapshots location would further improve the performance of the method. This improvement would be quite convenient since POD + GA calculations are quite fast compared with the CFD calculation of the snapshots, which is by far the slowest part of the process.

Table 1. RMS Errors in Percent in the 28 Test Points Resulting from Calculating the Six Aerodynamic Coefficients with the Three Combinations of Snapshots

Aerodynamic coefficients	Combination			
	1 O-Mesh	1 Reduced	2 Reduced	3 Reduced
C_L	0.39 (1.06)	0.40 (1.28)	1.40 (2.76)	2.18 (6.39)
C_{PD}	4.62 (12.8)	4.85 (14.5)	4.96 (11.1)	7.17 (18.0)
C_Y	2.01 (5.81)	2.05 (5.93)	2.34 (5.80)	2.97 (7.43)
C_M	0.30 (0.73)	0.28 (0.84)	1.00 (2.00)	1.47 (3.74)
C_R	0.37 (0.97)	0.39 (1.17)	1.32 (2.59)	2.07 (5.90)
C_N	4.12 (11.9)	4.31 (13.0)	2.09 (4.80)	4.10 (7.90)

Note: Maximum percent errors are also given in parentheses.

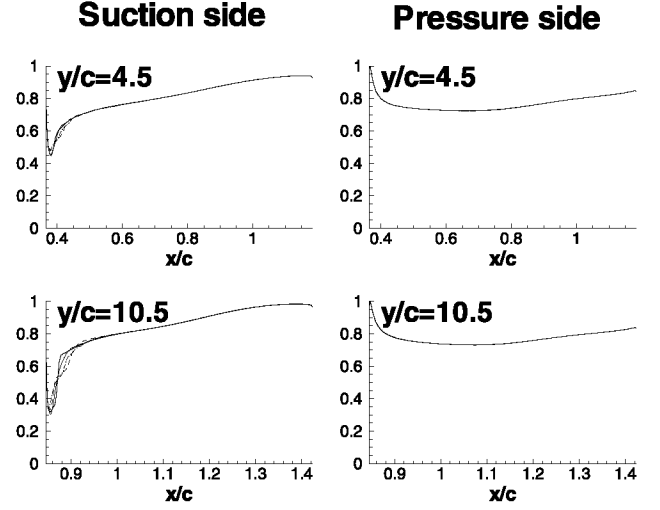


Fig. 3. Normalized pressure coefficient (C_p/C_{pmax}) along the indicated $y/c = \text{constant}$ lines at test point PT73: CFD (thick solid lines) and ROM, with Combinations 1 (thin solid), 2 (dot-dashed), and 3 (dashed)

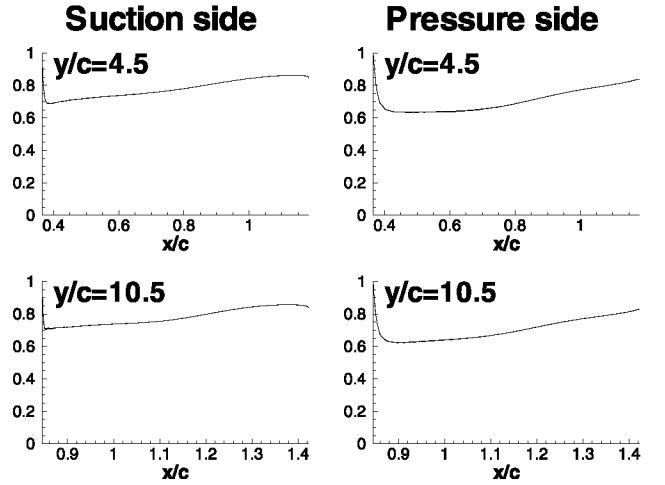


Fig. 4. As in Fig. 3, but considering test point PT81

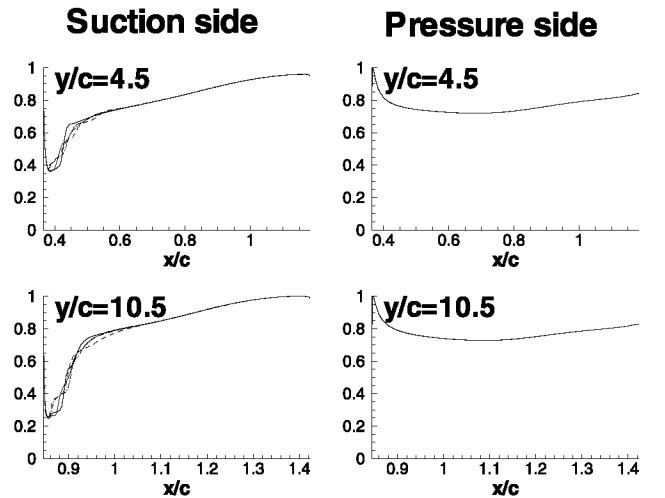


Fig. 5. As in Fig. 3, but considering test point PT84

The required numbers of modes that are selected in the local POD manifolds associated with the pressure and suction subdomains depend on the test point that is being reconstructed, varying between a minimum and maximum numbers, n_{\min} and n_{\max} . For the pressure side domain, $(n_{\min}^{PS}, n_{\max}^{PS}) = (3, 5)$, $(5, 6)$, and $(6, 7)$, for combinations 1, 2, and 3, respectively; for the suction side, the minimum and maximum numbers of modes are $(n_{\min}^{SS}, n_{\max}^{SS}) = (3, 19)$, $(9, 12)$, and $(11, 11)$, respectively. Note that the number of modes is generally larger in the suction side, which is because the flow acceleration is stronger in the suction side, which thus shows stronger shock waves. Also, the difference between n_{\min} and n_{\max} increases as the number of snapshots increases. In other words, as the quality of the POD manifold increases, the ability of the manifold to adapt itself to each reconstruction also increases.

The ability of the ROM to provide the local C_p distributions on the HTP surface is illustrated in Figs. 3–5, where the C_p distributions along the chord are given at two spanwise sections in the pressure and suction sides, for the test points PT73, PT81, and PT84, respectively. The required numbers of POD modes to reconstruct the pressure and suction sides, using Combinations 1, 2, and 3 are as follows:

- PT73: $(n^{PS}, n^{SS}) = (5, 17)$, $(6, 12)$, and $(7, 11)$, respectively.
- PT81: $(n^{PS}, n^{SS}) = (5, 6)$, $(6, 12)$, and $(7, 11)$, respectively.
- PT84: $(n^{PS}, n^{SS}) = (5, 19)$, $(6, 12)$, and $(7, 11)$, respectively.

Note that when invoking Fig. 2(a), these three test points are representative of the transonic regime, which is the most demanding region in the parameter space; in fact, the C_p -reconstructions (omitted here) at those test points for $M \leq 0.6$ are plot-indistinguishable from their CFD counterparts, even using Combination 3. To better appreciate the various approximations, each C_p distribution is normalized with its maximum along the chord $C_{p_{\max}}$. Results are quite good when no shock wave is present (e.g., plots in Fig. 4 and plots for the pressure side in Figs. 3 and 5), although some discrepancies appear near shock waves, where some stairlike structures can be appreciated that are similar to those encountered in the 2D case studied by Alonso et al. (2010). As expected, reducing the number of snapshots degrades the approximation and such degradation concentrates near the shock wave.

The advantages of defining a local POD is illustrated in Fig. 6, where the energy level of the various POD modes for the pressure [Fig 6(a)] and suction [Fig 6(b)] side regions is plotted versus the modes number, as resulting from application of both global POD (thick solid line) and local POD at the test points PT73 (thin solid line), PT81 (thin dashed line), and PT84 (thin dot-dashed line). As usually, the normalized energy of the i th mode is defined as

$$E_i = \frac{\gamma_i}{\sum_i \gamma_i}$$

where $\gamma_1, \gamma_2, \dots$ are the squares of the singular values [i.e., the eigenvalues of the covariance matrix Eq. (4)]. Note that the energy of local POD modes decreases faster than that of global modes, as expected. Such faster decrease is more evident in the pressure and suction sides at PT81 and in the pressure side at the remaining two points, that is, in those cases in which no shock wave is present (see Figs. 3–5). As already explained, the presence of shock waves that move as the parameters are varied is mainly responsible for the need to retain a large number of POD modes, which happens especially near the $(AoA, M) = (3^\circ, 0.8)$ corner of the parameter space, where the PT73 and PT84 points are located (see Fig. 2).

To compare (both in the O-mesh and in the whole computational domain) the reconstructions of the whole aerodynamic flow provided by the various combinations of snapshots, the RMS error in each flow variable for the 28 test points is now considered. The normalized RMS error is defined as

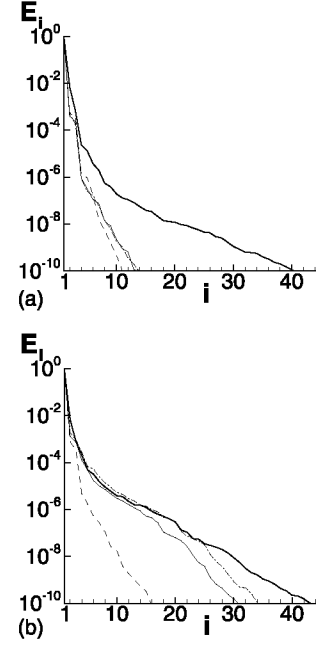


Fig. 6. Normalized energy level E_i versus the mode number i , resulting from applying global (thick line) and local (thin lines) POD at the test points considered in Figs. 3–5: (a) pressure side; (b) suction side

$$RMSE = \frac{1}{\tilde{A}} \sqrt{\sum_{k=1}^{28} \langle E_k, E_k \rangle} \quad (23)$$

Here, $\langle \cdot, \cdot \rangle$ is the inner product of Eq. (17), with the sum in the right side of Eq. (17) extended to all elementary cells in either the O-mesh or the whole computational domain; the normalizing area \tilde{A} is defined as

$$\tilde{A} = \sqrt{\sum_{k=1}^{N_R} \tilde{A}_k^2}$$

where $\tilde{A}_1, \dots, \tilde{A}_{N_R}$ are the areas of the elementary cells that are being considered; the pointwise error appearing in Eq. (23) is defined for each flow variable ϕ as

$$E(x, y, z) = \frac{|\phi_{\text{CFD}}(x, y, z) - \tilde{\phi}_{\text{ROM}}(x, y, z)|}{\max |\phi_{\text{CFD}}|} \quad (24)$$

Here, the subscripts CFD and ROM refer to the CFD and ROM approximations, respectively, and $\max |\phi_{\text{CFD}}|$ denotes the maximum, absolute value of the flow variable in Ω , where Ω is either the O-mesh or the whole computational domain.

Results on the various flow variables distributions are given in terms of RMS errors in Table 2, where errors smaller than 0.01% are rounded off to zero. Maximum errors are not shown because they do not illustrate how good the approximate solution is, since small errors in the position of the shock waves lead to large local errors. Table 2 shows that

1. The RMS errors in the O-mesh are always larger than their counterparts in the whole computational domain, which was to be expected. This is because the O-mesh includes the richest flow structure and exhibits a quite small volume compared with that of the whole computational mesh.

Table 2. Normalized [as Defined in Eq. (23)] RMS Errors in Percent in the O-Mesh of the Flow Variables for the Three Combinations of Snapshots

Flow variables	Combination		
	1	2	3
ρu	0.38(0.01)	0.60(0.03)	1.02(0.03)
ρv	0.40(0.00)	0.60(0.00)	0.93(0.00)
ρw	0.18(0.14)	0.22(0.25)	0.31(0.27)
ρ	0.23(0.01)	0.30(0.02)	0.40(0.02)
p	0.25(0.01)	0.33(0.02)	0.42(0.02)

Note: Normalized RMS errors in the whole computational mesh are given in parentheses.

2. The larger the number of snapshots, the better the results, as expected.
3. The results are quite good, even when using the coarsest Combination 3, with only 25 snapshots. This produces RMS errors that are less than 1%, which is more than enough in most industrial applications.

Notice that the aerodynamic field in the whole computational mesh (including the wake and the boundary layer) has also been reconstructed quite well using only a limited number of mesh points that are located in the O-mesh:

- The approximation outside the O-mesh (including the wake) is illustrated in Fig. 7, where the ρu and ρw distributions along the lines $y = 0, z = 0, x = c/4, z = 0$, and $x = c/4, y = 0$ are shown for the test point PT84 (which is the test point that shows largest errors in Table 2); as in Figs. 3–5, CFD and ROM results using Combinations 1, 2, and 3 are plotted with thick solid, thin solid, thin dot-dashed, and thin dashed lines, respectively. The gaps appearing in this figure correspond to the intersection of the HTP with the considered lines. Note that the approximation is quite good with the three combinations, except for a small discrepancy in the wake when using combination 3, and a quite small discrepancy in the far field for the flow variable ρu , which is due to the fact that the free stream boundary conditions are not exactly satisfied; the latter discrepancy could be avoided using some additional mesh points in the far field to define the residual.
- The approximation in the layer is illustrated for the test point PT84 in Fig. 8, where the x -mass flux is plotted versus the wall-normal coordinate at four representative points on the HTP surface; as in Figs. 3–7, CFD and ROM results using Combinations 1, 2, and 3 are plotted with thick solid, thin solid, thin dot-dashed, and thin dashed lines, respectively. Note that the results are quite good (CFD and ROM results are almost indistinguishable) in three of the points. This is because these three points are not in a vicinity of a shock wave. The fact that these results are so good could be seen as surprising at first sight because ROM calculations were based on the Euler equations, which do not apply in the boundary layer; and furthermore, the residual has been calculated using only information from the projection window, which is located outside the boundary layer. The reason is that the boundary layer structure is somewhat slaved (along the wall-normal direction) to the outer flow. Thus, if the latter is well calculated (which occurs if no nearby shock wave is present), then the boundary layer must be well approximated too, since the relation between both is already present in the CFD calculated snapshots. When the point is under a shock wave, the approximation degrades.

Summarizing, the ROM developed previously provides quite good approximations of (1) the HTP aerodynamic coefficients (Table 1), (2) the C_p distribution along the HTP surface (Figs. 3–5), and (3) the aerodynamic field in the whole computational (Table 2 and Fig. 7) domain, including the wake (Fig. 7) and the boundary

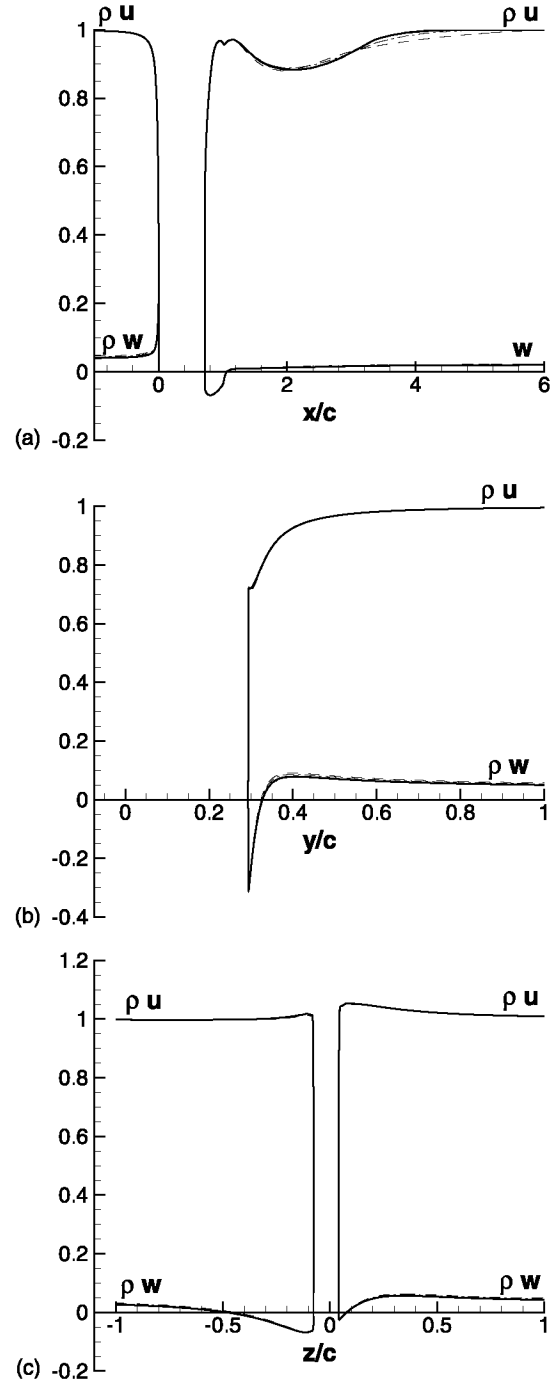


Fig. 7. Approximation outside the O-mesh: distribution of ρu and ρw along the lines (a) $y = 0, z = 0$, (b) $x = 0.25 \cdot c, z = 0$, and (c) $x = 0.25 \cdot c, y = 0$ at the test point PT84

layer (Fig. 8). These precise results were caused in part by the ability of the ROM to avoid localized CFD errors, which were excluded from the projection window. Note that precision was maintained in spite of the various simplifications that have been introduced. In particular, the residual was based on the Euler equations, and both the covariance matrix and the residual were calculated using only 1,664 points (defining 208 elementary cells) scattered on a part of an O-mesh in the vicinity of the HTP, instead of the total number of mesh points in the computational window (i.e., three million points). These simplifications greatly reduced the computational effort. In fact, even though the residual was minimized using a genetic

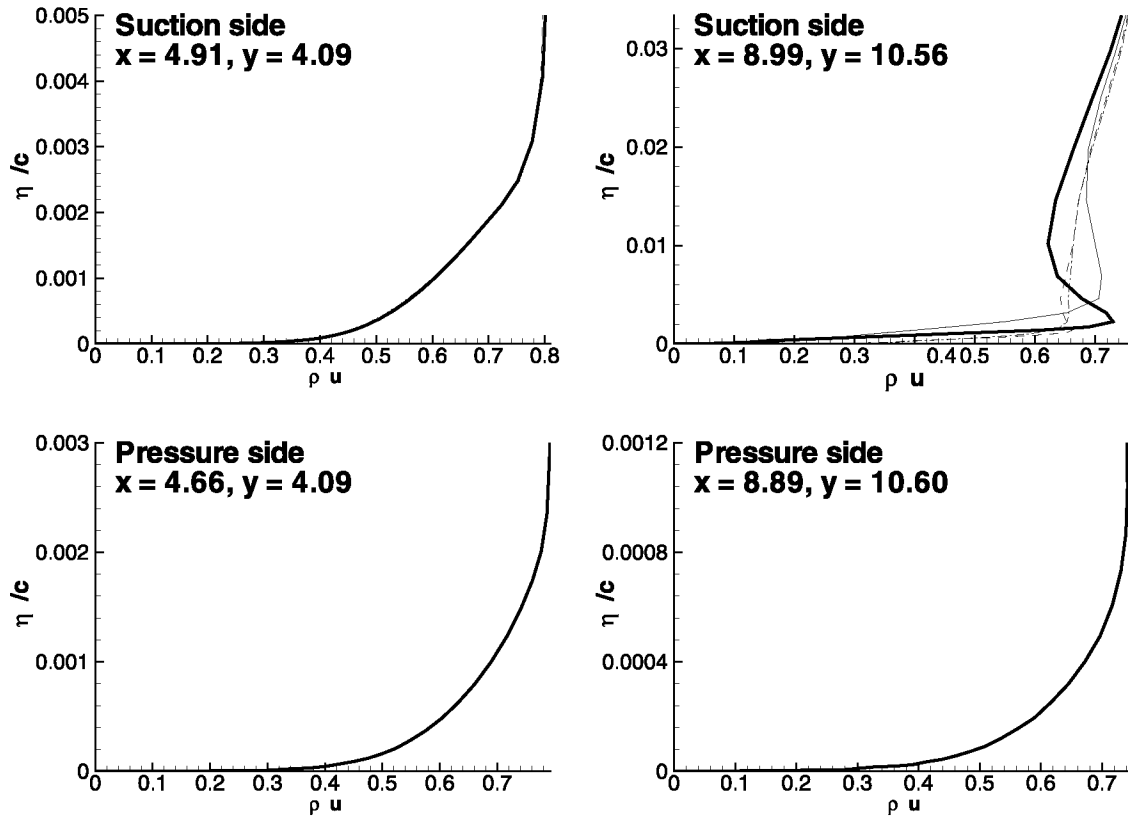


Fig. 8. Boundary layer: streamwise mass flux ρu distribution along the wall-normal coordinate at four points on the HTP surface (as indicated) for the test point PT84

algorithm (a fairly slow method), the ROM needed only 7 CPU minutes in a standard PC to reconstruct each flow configuration in the O-mesh (which is more than enough to calculate aerodynamic loads) using 117 snapshots. Reconstruction of the flow in the whole computational domain requires 10 additional CPU minutes using 117 snapshots. These CPU times compare quite well with the 3 CPU days that are required by the CFD tool. It is also to be noted that a somewhat crude software (based on *FORTRAN90*) has been used and that all ROM calculations are quite suitable for parallelization.

Concluding Remarks

A method has been developed to construct ROMs of 3D, steady aerodynamic flow fields of industrial interest depending on several parameters. The method is based on the calculation of various aerodynamic fields, called the snapshots, for representative values of the parameters. The CFD solver is a typical RANS solver used in the aeronautic industry. The ROM is constructed minimizing a properly defined residual based on the governing equations and boundary conditions. The method has been tested considering the aerodynamic field around a commercial aircraft HTP, depending on two parameters, the Mach number and the angle of attack, in a range that included transonic flows. Various ingredients have been included in the ROM that (to our knowledge) are essentially new in this context and improve the computational efficiency:

- It is the conservative Euler equations and not the original modified NS equations plus turbulence model that are used to calculate the residual. This makes the method independent of the turbulence model/numerical stabilizers that might have been used, which is convenient in industrial applications.

- Both the residual and the covariance matrix of the snapshots set are calculated using a limited number of mesh points only that are concentrated in a projection window. This strongly improves computational efficiency maintaining precision, provided that the projection window includes the relevant aerodynamic information. Localized CFD errors can be avoided excluding them from the projection window.
- The POD manifold can be defined locally in the parameter space, and calculations can be made in various subdomains of the projections window, which makes the method quite flexible. Various additional improvements can be envisaged, which are the object of our current/planned research:
- Location of the snapshots in the parameter space is very important, which suggests that a method to select the snapshots position would further increase the computational efficiency.
- The computational cost would be further reduced replacing the slow GA method that has been used by faster gradient-like methods.
- Control surfaces with varying deflection angles can be accounted for considering virtual meshes to which the various deflected meshes are projected upon. This can be done using ideas similar to those developed by Bache et al. (2010) in a related low Reynolds number problem that is of interest in microfluidics.
- Larger values of the Mach number and/or the angle of attack may produce shock wave structures that move significantly as the parameters are varied. The subsequent increase in the required number of snapshots could be avoided extending to 3D the shock wave treatment method developed in 2D (Alonso et al. 2010).

The ROM developed previously has been designed keeping in mind its industrial use, to improve the somewhat rough, ad hoc methods that are used nowadays in industry to cope with multiparameter aerodynamic flows. We hope that this will be a step further

in current efforts to speed up activities on aerodynamic design and certification using new computational tools.

Acknowledgments

This research has been partially supported by Airbus Operations under Contract No. A8208636G. The work by José M. Vega has also been supported by the Spanish Ministry of Education, under Grant No. TRA2010-18054. We are indebted to two anonymous referees for some useful comments on an previous version of the paper.

Notation

The following symbols are used in this paper:

- \mathcal{A} = area of the surface Γ ;
- \bar{A} = normalized area;
- A_j = POD-mode amplitude of the j th POD mode;
- AoA = angle of attack;
- BC_i = i th boundary condition;
- C_L = lift coefficient;
- C_M = pitch moment coefficient;
- C_N = yaw moment coefficient;
- C_P = pressure coefficient;
- C_{PD} = pressure drag coefficient;
- C_R = roll moment coefficient;
- C_Y = lateral force coefficient;
- c = root chord;
- D_j = distance from each snapshot to \tilde{q} ;
- E_i = normalized energy of the i th mode;
- EQ_i = i th equation;
- e = modified mechanical energy per unit mass;
- F = safety factor;
- \mathcal{H} = residual;
- M = Mach number;
- \tilde{N} = selected number of snapshots;
- N_E = number of cells in the domain;
- N_0 = total number of snapshots;
- n = truncation number;
- \mathbf{n} = outward y -unit normal to the surface Γ ;
- (n_x, n_y, n_z) = (x, y, z) -component of \mathbf{n} ;
- p = pressure;
- Q_i = i th POD mode;
- q_i = i th snapshot;
- $\tilde{\mathbf{q}}$ = state vector;
- \mathbf{R} = covariance matrix;
- S = surface area;
- T = temperature;
- (u, v, w) = (x, y, z) -velocity component;
- α_j^k = k th component of the j th eigenvector of the covariance matrix;
- Γ = boundary of domain;
- γ = specific heats ratio;
- γ_i = i th eigenvalue of the covariance matrix;
- ε = error bound;
- ϕ = flow variable;
- ρ = density; and
- Ω = domain.

References

- Akima, H. (1978). "A method of bivariate interpolation and smooth surface fitting for irregular distributed data points." *ACM Trans. Math. Softw.*, 4(2), 148–159.
- Alonso, D., Vega, J. M., and Velazquez, A. (2010). "Reduced-order model for viscous aerodynamic flow past an airfoil." *AIAA J.*, 48(9), 1946–1958.
- Alonso, D., Velazquez, A., and Vega, J. M. (2009). "A method to generate computationally efficient reduced order models." *Comput. Methods Appl. Mech. Eng.*, 198(33–36), 2683–2691.
- Bache, E., Vega, J. M., and Velazquez, A. (2010). "Model reduction in the back step fluid–thermal problem with variable geometry." *Int. J. Therm. Sci.*, 49(12), 2376–2384.
- Ben Kheli, S., Gervois, J. L., Carrier, G., Moens, F., and Viscat, P. (2002). "Assessment of ELSA software through civil transport aircraft configurations." *Proc., EAS Aerospace Aerodynamics Research Conf.*, Engineering and Applied Science, Cambridge, U.K.
- Cambier, L. and Gazaix, M. (2002). "ELSA: An efficient object-oriented solution to CFD complexity." *40th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Reston, VA.
- Couplet, M., Basdevant, C., and Sagaut, P. (2005). "Calibrated reduced-order POD–Galerkin system for fluid flow modelling." *J. Comput. Phys.*, 207(1), 192–220.
- Dowell, E. H., and Hall, K. C. (2001). "Modeling of fluid–structure interaction." *Annu. Rev. Fluid Mech.*, 33, 445–490.
- Edwards, J. R., and Chandra, S. (1996). "Comparison of eddy viscosity–transport turbulence models for three-dimensional, shock-separated flow fields." *AIAA J.*, 34(4), 756–763.
- LeGresley, P. A., and Alonso, J. J. (2001). "Investigation of non-linear projection for POD based reduced order models for aerodynamics." *39th AIAA Aerospace Sciences Meeting & Exhibit*, American Institute of Aeronautics and Astronautics, Reston, VA.
- Lieu, T., Farhat, C., and Lesoinne, M. (2006). "Reduced-order fluid/structure modeling of a complete aircraft configuration." *Comput. Methods Appl. Mech. Eng.*, 195(41–43), 5730–5742.
- Lorente, L. (2009). "Generation and post-process of aerodynamic databases via SVD/POD methods." Ph.D. thesis, Escuela Tecnica Superior de Ingenieros Aeronauticos, Universidad Politecnica de Madrid, Madrid, Spain.
- Lucia, D. J., Beran, P. S., and Silva, W. A. (2004). "Reduced-order modeling: New approaches for computational physics." *Prog. Aerosp. Sci.*, 40(1–2), 51–117.
- Rapun, M. L., and Vega, J. M. (2010). "Reduced order models based on local POD plus Galerkin projection." *J. Comput. Phys.*, 229(8), 3046–3063.
- Rempfer, D. (2003). "Low dimensional modeling and numerical simulation of transition in a simple shear flow." *Annu. Rev. Fluid Mech.*, 35, 229–265.
- Shepard, D. (1968). "A two-dimensional interpolation function for irregularly-spaced data." *Proc., 1968 ACM National Conf.*, Association for Computing Machinery, New York.
- Sirisup, S., and Karniadakis, G. E. (2005). "Stability and accuracy of periodic flow solutions obtained by POD-penalty method." *Physica D*, 202(3–4), 218–237.
- Sirisup, S., Karniadakis, G. E., Xiu, D., and Kevrekidis, I. G. (2005). "Equation-free/Galerkin-free POD-assisted computation of incompressible flows." *J. Comput. Phys.*, 207(2), 568–587.
- Spalart, P. R., and Allmaras, S. R. (1992). "A one-equation turbulence model for aerodynamic flows." *30th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, VA.
- Tannehill, J. C., Anderson, D. A., and Pletcher, R. H. (1997). *Computational fluid mechanics and heat transfer*, Taylor & Francis, Philadelphia.
- Thomas, J. P., Dowell, E. H., and Hall, K. C. (2010). "Using automatic differentiation to create nonlinear reduced-order-model aerodynamic solver." *AIAA J.*, 48(1), 19–24.